Middle Eastern Finance and Economics
ISSN: 1450-2889 Issue 6 (2010)

http://www.eurojournals.com/MEFE.htm

Using Technical Analysis with Neural Network for Forecasting Stock Price Index in Tehran Stock Exchange

Mohsen Mehrara
Faculty of Economics, University of Tehran, Tehran, Iran
E-mail: mmehrara@ut.ac.ir

Ali Moeini
Department of Algorithms and Computation, Faculty of Engineering
University of Tehran, Tehran, Iran
E-mail: moeini@ut.ac.ir

Mehdi Ahrari
Faculty of Economics, University of Tehran, Tehran, Iran
E-mail: ahrari@ut.ac.ir

Alireza Ghafari
Faculty of Economics, University of Tehran, Tehran, Iran
E-mail: ghafari@ut.ac.ir

Abstract

Neural networks have the advantage of simulating the non-linear models when little a priori knowledge of the structure of problem exist or the number of immeasurable input variables are great and system has a chaotic characteristic. Among different methods, MLFF neural network with back-propagation learning algorithm and GMDH neural network with Genetic algorithm (GA) learning are used to predict TEPIX based on the TSE database. This paper uses moving average crossover inputs based on technical analysis rules and the results show the exponential moving average has better result than simple moving average and also the GMDH has better result in the forecasting, power tracking and profitability relative to MLFF neural network.

Keywords: Multi-Layered Feed Forward (MLFF), Group Method of Data Handling (GMDH), Technical Analysis, Tehran Exchange Price Index (TEPIX), Tehran Stock Exchange (TSE).

Introduction

The Tehran Stock Exchange began dealing in the shares of a few private banks and companies, as well as treasury bonds and state-backed securities, in 1968. During the 1970s, and particularly after the 1974 oil price explosion, the market showed a dramatic rise in volume of transactions with modest share price rise. In 1977-78, the annual value of shares traded in the TSE reached IR44.5 billion ($628mn) – a record for the decade. On the eve of the 1979 revolution the paid-up capital of 105 enterprises (22 private banks, 2 insurance companies, and 81 industrial corporations) listed on the exchange was estimated at IR220 billion ($3.1billion)

By the 1979 Islamic revolution, 105 firms were listed on the exchange. That number fell to 56 after the revolution, as private banks were nationalized and enterprises belonging to the royal family were expropriated. Islamic regulations against interest payments, and the 1980-88 Iran-Iraq War all stifled activity on the stock exchange. Annual volume of trades fluctuated between a miniscule IR49mn in 1981-82 and IR1, 690mn in 1987-88 – mostly involving equities of less than 40 companies.

"Privatization program," launched under the first five-year plan (Five-Year Economic Development Plan) in 1991-92, to sell some 390 state firms to the public became a welcome boon to the exchange's upswing. Under this initiative post-revolution public entities and several Islamic "charitable" organizations (the institutes) that had previously taken over the expropriated private properties were ordered by the government to offer shares of their enterprises to the public through the TSE. As a result, the total number of shares traded in that year reached 62.6mn worth IR478bn, and the TEPIX registered 472 (1990=100) – a post-revolution record. Allegations of questionable accounting procedures, fabricated corporate profits before their initial public offerings (IPO), and other unsavory practices in the sale of public companies caused the TEPIX to decline by 15% in two subsequent years – taking the gloss off the nascent market.

The exchange enjoyed a brief surge from 1994 through 1997 before tapering off. When the annual money supply increased and there was a mild recession in other prospective areas of investment, there was a "meteoric boom." The Privatization Agency's initial public offerings (IPOs) contributed to this. In the first six months of 2002-03, the TEPIX rose from 5,368 to 8,993, and in the next six months to 11,379. Thus, in a span of only four years between March 1999 and March 2003, the index rose from about 2,206 to nearly 11,400, and the number of shares traded in the market went up from 1.7billion a year to 7.9 billion. The market continued its upward trend and reached an all time record high of 13,836 in mid-December 2004. Capitalization: $36,440 milliard (2004) since 2005, foreign investors have been able to participate in the TSE. Foreign investors are permitted to hold a maximum of 10% of shares of companies. Trading volume: $ 7,866 milliard (2005) at the end of 2007, TSE market capitalization stood at $46 billion. The stock market capitalization of listed companies in Iran was valued at $70 billion in 2008.

Stock markets are affected by many highly interrelated economic, political, and sentimental factors, which often interact with one another in a very complex manner. As such, it has always been very difficult to forecast the movements of stock prices and market indices. The methods that had been used to predict market prices fall broadly into three categories fundamental analysis, technical analysis and traditional time series forecasting. Fundamental analysis (Ritchie [1]) presumes that the price of a stock depends on its intrinsic value and anticipated return on investment. it is possible to determine the company's intrinsic value and expected returns By analyzing the company's operations and the market in which the company is operating. Fama and Schwert[2] Rozeff[3] Keim and Stambaugh[4]. Most people believe that for short- and medium-term speculations, fundamental analysis is generally not appropriate.

Technical analysis (Murphy [5]) refers to the various methods that aim to predict future price movements using past stock prices and volume information. It is based on the assumption that history repeats itself and that future market directions can be determined by examining historical price data. Most of the techniques used in technical analysis are highly subjective in nature and have been shown not to be statistically valid. (coulson [6] Van eyden [7]).

Traditional time series forecasting (Box and Jenkins [8]) techniques in statistics have also been applied to predicting stock price movements. Using techniques such as autoregressive integrated moving-average (ARIMA) or multivariate regression; it is possible to model historical price data as a nonlinear function using a recurrence relation. A good example of using multivariate regression to predict the S&P 500 index and the Dow Jones Industrial Average (DJIA) is presented by Pesaran and Timmermann [9].

Neural network model is an emerging computational technology that provides a new avenue for exploring the dynamics of various economic and financial applications. Forecasting stock prices a good example to demonstrate the power of various Artificial intelligence techniques in teaching and learning. A NN is an interconnected network of simple processing elements (artificial neurons) with a different weight associated with each connection. With a proper network topology and appropriate weights between the connections, a NN can be trained to approximate any function mapping between its input(s)and output(s) by using an appropriate learning algorithm such as back propagation (BP) (Rumel hart et al.a,b [10]).

Artificial neural networks are an information processing technology for modeling mathematical relationships between input and output variables. In recent years, artificial neural networks (NNs) (Haykin [11]) have become another important technique for predicting stock prices. Researches have used ANNs on the economic applications is expanding rapidly (Meraviglia[12] Shachmurove[13], Zhang & Berardi[14]). Recently some studies have empirically forecasted macroeconomic variables such as inflation, interest rates and exchange rate (Bissoondeeal, Binner, Elger, Gazely, & Mullineux[15], Plasmans, Verkooijen , & Daniels[16] Qi & Wu[17],Saltoglu[18]). This approach is effective for input and output relationship modeling even for noisy data and has been demonstrated to effectively model nonlinear relationships.Such as, related studies have estimated and forecasted stock prices (Donaldson & Kamstra[19], Black & McMillan[20] Jasic & Wood [21] Kanas & Yannopoulos1[22] Kanas[23], Maasoumi & Racine[24] Qi[25] Rapach & Wohar[26] ,Shively[27]) and stock volatilities (Hamid & Iqbal [28] Dunis & Huang [29]). There have been number of attempts to apply ann to the task of modeling security prices(lam[30],nygren[31] kaastra and boyd [32]jasic and wood[33]) Moreover, major studies on derivative securitiespricing using neural network have attracted researchers and practitioners, and they applied the neural network model and obtained better results than using the traditional option-pricing model (Yao, Li, & Tan[34], Amilon[35], Malliaris & Salchenberger [36], Binner et al [37], Heston & Nandi[38], Hutchinson, Lo, & Poggio [39], Lin & Yeh[40] Qi[41]) A formal study of the predictive power of technical analysis rules in linear and nonlinear models is given by Neftci [42] Brock et al. [43] provide some of the earliest support for the use of technical analysis and suggest the need for a nonlinear model. Gencay [44] use foreign exchange markets to pioneer the use of technical analysis rules as inputs for neural networks, which are flexible, nonlinear models with powerful pattern recognition properties. In a series of articles, Gencay [45] and Gencay [46] and Gencay and Stengos [47] show that simple technical rules result in significant forecast improvements for current returns over a random walk model for both foreign exchange rates and stock indices.

In this paper, a method of non-parametric approach to predict the TEPIX over time in different condition of market is used. MLFF neural network with back-propagation algorithm and GMDH neural network with GA are briefly discussed, and they are used to make appropriate model for predicting the TEPIX of the TSE database. As input variables to the neural networks, we use price time series separately with 2 lags of the 5, 10, 50 and 55-day moving average crossover. These two models are coded and implemented in Matlab software package. The prime aim here is to find out dependence in TEPIX and also which neural network model does best in forecasting when the input parameters are little or great.

The remainder of the paper is organized as follows. Section 2 describes the non-parametric modeling approach adopted here. Section 3 discusses relevant concepts of the technical indicators used for the study. Empirical results are presented in Section 4, and concluding remarks in Section 5.

2. Modeling using Neural Network

Artificial Neural Networks (ANN) is biologically inspired network based on the organization of neurons and decision making process in the human brain. In other words, it is the mathematical analogue of the human nervous system. This can be used for prediction, pattern recognition and pattern classification purposes. It has been proved by several authors that ANN can be of great use when the associated system is so complex that the underline processes or relationship are not completely understandable or display chaotic properties. Development of ANN model for any system involves three important issues: (i) topology of the network, (ii) a proper training algorithm and (iii) transfer function. Basically an ANN involves an input layer and an output layer connected through one or more hidden layers. The network learns by adjusting the inter connections between the layers. When the learning or training procedure is completed, a suitable output is produced at the output layer. The learning procedure may be supervised or unsupervised. In prediction problem supervised learning is adopted where a desired output is assigned to network before hand. Based on research aim, varieties of artificial neural networks are used

2.1. MLFF Neural Network

MLFF neural network is one the famous and it is used at more than 50 percent of researches that are doing in financial and economy field recently [48]. This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as a transfer function ($f(x)=\frac{1}{1+e^{-x}}$). It has a continuous derivative, which allows it to be used in back-propagation. This function is also preferred because its derivative is easily calculated: $y' = y(1-y)$

Multi-layer networks use a variety of learning techniques; the most popular is back-propagation algorithm (BPA). The BPA is a supervised learning algorithm that aims at reducing overall system error to a minimum [49]. This algorithm has made multilayer neural networks suitable for various prediction problems. In this learning procedure, an initial weight vectors w_0 is updated according to [50]:

$$w_i(k+1) = w_i(k) + \mu(T_i - O_i) f'(w_i x_i) x_i \qquad \text{(a)}$$

Where, $w_i \Rightarrow$ the weight matrix associated with ith neuron; $x_i \Rightarrow$ Input of the ith neuron; $O_i \Rightarrow$ Actual output of the ith neuron; $T_i \Rightarrow$ Target output of the ith neuron, and μ is the learning rate parameter.

Here the output values (O_i) are compared with the correct answer to compute the value of some predefined error-function. The neural network is learned with the weight update equation (a) to minimize the mean squared error given by [51]:

$$E = \frac{1}{2}(T_i - O_i)^2 = \frac{1}{2}[T_i - f(w_i x_i)]^2$$

By various techniques the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculations is small. In this case one says that the network has learned a certain target function. To adjust weights properly one applies a general method for non-linear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases [52].

The gradient descent back-propagation learning algorithm is based on minimizing the mean square error. An alternate approach to gradient descent is the exponentiated gradient descent algorithm which minimizes the relative entropy [53].

2.2. GMDH Neural Network

By means of GMDH algorithm a model can be represented as sets of neurons in which different pairs of them in each layer are connected through a quadratic polynomial and thus produce new neurons in the next layer. Such representation can be used in modeling to map inputs to outputs [54, 55,56]. The formal definition of the identification problem is to find a function $\hat{f}$ so that can be approximately used instead of actual one, f, in order to predict output $\hat{y}$ for a given input vector $X = (x_1, x_2, x_3, ..., x_n)$ as close as possible to its actual output y. Therefore, given M observation of multi-input-single-output data pairs so that

$$y_i = f(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) \quad (i=1,2,\ldots,M)$$

it is now possible to train a GMDH-type neural network to predict the output values $\hat{y}_i$ for any given input vector $X = (x_{i1}, x_{i2}, x_{i3}, ..., x_{in})$, that is

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) \quad (i=1,2,\ldots,M).$$

The problem is now to determine a GMDH-type neural network so that the square of difference between the actual output and the predicted one is minimized, that is

$$\sum_{i=1}^{M} [\hat{f}(x_{i1}, x_{i2}, x_{i3}, ..., x_{in}) - y_i]^2 \to \min .$$

General connection between inputs and output variables can be expressed by a complicated discrete form of the Volterra functional series in the form of

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \ldots \quad (1)$$

which is known as the Kolmogorov-Gabor polynomial [54, 55, 57]. This full form of mathematical description can be represented by a system of partial quadratic polynomials consisting of only two variables (neurons) in the form of

$$\hat{y} = G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \quad (2)$$

In this way, such partial quadratic description is recursively used in a network of connected neurons to build the general mathematical relation of inputs and output variables given in equation (1). The coefficient a_i in equation (2) are calculated using regression techniques so that the difference between actual output, y, and the calculated one, $\hat{y}$, for each pair of x_i, x_j as input variables is minimized. Indeed, it can be seen that a tree of polynomials is constructed using the quadratic form given in equation (2) whose coefficients are obtained in a least-squares sense. In this way, the coefficients of each quadratic function G_i are obtained to optimally fit the output in the whole set of input-output data pair, that is:

$$E = \frac{\sum_{i=1}^{M} (y_i - G_i)^2}{M} \to \min \quad (3)$$

In the basic form of the GMDH algorithm, all the possibilities of two independent variables out of total n input variables are taken in order to construct the regression polynomial in the form of equation (2) that best fits the dependent observations (y_i, i=1, 2, …, M) in a least-squares sense. Consequently, $\binom{n}{2} = \frac{n(n-1)}{2}$ neurons will be built up in the first hidden layer of the feed forward network from the observations $\{(y_i, x_{ip}, x_{iq});$ (i=1, 2,… M)$\}$ for different $p, q \in \{1,2,...,n\}$. In other words, it is now possible to construct M data triples $\{(y_i, x_{ip}, x_{iq});\ (i=1, 2,\ldots, M)\}$ from observation using such $p, q \in \{1,2,\ldots,n\}$ in the form

$$\left[\begin{array}{cc|c} x_{1p} & x_{1q} & y_1 \\ x_{2p} & x_{2q} & y_2 \\ \hline x_{Mp} & x_{Mq} & y_M \end{array}\right].$$

Using the quadratic sub-expression in the form of equation (2) for each row of M data triples, the following matrix equation can be readily obtained as $A\mathbf{a} = Y$

where $\mathbf{a}$ is the vector of unknown coefficients of the quadratic polynomial in equation (2)

$$\mathbf{a} = \{a_0, a_1, a_2, a_3, a_4, a_5\} \tag{4}$$

and $Y = \{y_1, y_2, y_3, \ldots, y_M\}^T$ is the vector of output's value from observation. It can be readily seen that

$$A = \left[\begin{array}{cccccc} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ \hline 1 & x_{Mp} & x_{Mq} & x_{Mp}x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \end{array}\right]$$

The least-squares technique from multiple-regression analysis leads to the solution of the normal equations in the form of

$$\mathbf{a} = (A^T A)^{-1} A^T Y \tag{5}$$

which determines the vector of the best coefficients of the quadratic equation (2) for the whole set of M data triples. It should be noted that this procedure is repeated for each neuron of the next hidden layer according to the connectivity topology of the network. However, such a solution directly from normal equations is rather susceptible to round off errors and, more importantly, to the singularity of these equations.

3. Technical Indicators used for the study

In this section, we introduce technical indicators that will have used in this research. So in Section 3.1 the moving average and in Section 3.2 tracking measurement technique are reviewed. Afterwards Section 3.3 reviews profitability measure.

3.1. Moving Average

Simple Moving Average (SMA): This is perhaps the oldest and the most widely used technical indicator. It shows the average value of price over time. A simple moving average with the time period n is calculated by:

$$SMA(t) = \frac{1}{n}\sum_{i=0}^{i=n} C_{t-i}$$

where C_t is a price at time t [58]. The shorter the time period, the more reactionary a moving average becomes. A typical short term moving average ranges from 5 to 25 days, an intermediate-term from 5 to 100, and long-term 100 to 250 days. In our experiment, the window of the time interval n is 5.

Exponential Moving Average (EMA): An exponential moving average gives more weight to recent prices, and is calculated by applying a percentage of today's closing price to yesterday's moving average. The equation for n time period is [58]:

$$EMA(t) = \frac{2}{n+1}C_t + \left(1 - \frac{2}{n+1}\right) * SMA(t)$$

The longer the period of the exponential moving average, the less total weight is applied to the most recent price. The advantage to an exponential average is its ability to pick up on price changes more. In our experiment, the window of the time interval n is 5.

3.2. Tracking Measurement Technique

After modeling and predicting of the TEPIX by using MLFF & GMDH neural networks, we have achieved following vectors for each method separately and used them to evaluate power tracking of above mentioned methods.

- First we made actual signal vector 'Y' as per followings:

$$\text{Signal Vector Y} \quad \begin{cases} \text{If } y_t - y_{t-1} \geq 0 \text{ then } Y=1 \\ \text{Else } Y=-1 \end{cases}$$

- Then predicted signal vector 'Ŷ' has been made:

$$\text{Signal Vector } \hat{Y} \quad \begin{cases} \text{If } \hat{y}_t - \hat{y}_{t-1} \geq 0 \text{ then } \hat{Y}=1 \\ \text{Else } \hat{Y}=-1 \end{cases}$$

- Next we have made tracking vector by multiplying actual & predicted signal vectors.

Tracking Vector A:

$$\begin{matrix} \text{If Y} \\ \text{If Y.} \end{matrix} \begin{cases} \hat{Y}=1 \text{ Then A=1 (Correct)} \\ \hat{Y}=1 \text{ Then A=0 (False)} \end{cases}$$

- So, we find out that the total number of correct tracking is equal to: $B=\sum A_i$

Ultimately, we calculated percentage of correct tracking based on number of prediction as per followings:

Percent of Correct Tracking = B / (Number of Predictions)*100

3.3. Profitability Measure

Moving average crossovers have long been used as buy–sell signals for trend-following trading systems. The assumption that a short-term moving average is higher than a long-term moving average indicates that prices are trending higher, thus signaling an uptrend. The opposite case signals a downtrend.

After forecast of the TEPIX by using MLFF & GMDH Neural Networks with moving average crossover inputs, a long position is taken when prices are predicted to be higher, and a short position when prices are predicted to be lower. Then profitability is calculated from actual returns based on these positions. Table 1 summarizes the trading strategies.

Table 1: Summary of trading approach

Approach	**Description**	**Strategy**
MLFF, GMDH	Inputs are the differences between the 5 ,50[MA_5 - MA_{50}], [$MA_5(-1)$ - $MA_{50}(-1)$], [$MA_5(-2)$ - $MA_{50}(-2)$] and the 10,55[MA_{10} - MA_{55}], [$MA_{10}(-1)$ - $MA_{55}(-1)$], [$MA_{10}(-2)$ - $MA_{55}(-2)$]-day moving average crossover of price (1–2 lags)	Take a long position if forecast from the network is that price will increase; take a short position if forecast is that price will decrease.

4. Empirical Results

In this Section, we first compare the results for two neural network whit moving averages used in this research in Section 4.1 and then compare power tracking of GMDH and MLFF approaches based on number of correct tracking for prediction sets in section 4.2. Afterwards, the comparison of profitability of two models is presented in Section 4.3.

4.1. Comparison of Moving Averages

As input variables to the neural networks, we use TEPIX time series covering 01/02/2002-01/10/2009 period separately, with 2 lags of the 5[MA_5,MA_5(-1),MA_5(-2)], 10[MA_{10},MA_{10}(-1),MA_{10}(-2)], 50[MA_{50},MA_{50}(-1),MA_{50}(-2)] and 55[MA_{55},MA_{55}(-1),MA_{55}(-2)] -day moving average crossover. 5 and 10-day simple moving average and EMA are used to measure the short-term dependency, and 50 and 55-day moving average are used to measure the long-term dependency in TEPIX.

Training set is determined to include about 50% of the data set and the 25% will be used for testing and 25% for validating purposes.

During the designing of MLFF model we have changed the number of layers and neurons. Finally we found the best network in this research as a network with two hidden layers and 14-7 neurons. Almost 100 transfer function has been used and sigmoid is used for the input function and the linear function is used for output. Table 2 summarizes the characteristics of the best network.

Table 2: MLFF network characteristics

NH	N	TF1	TF2	TFO
2	14 -7	Logistic	Tanh	Linear

Notes: NH stands for number of hidden layers; N for neurons in hidden layers; TFI for transfer function in hidden layer one; TF2 for transfer function in hidden layer two and TFO for transfer function in output layer;

The same way, the GMDH neural network model characteristics is summarized in Table 3.

Table 3: GMDH network characteristics

NH	N	TF1	TF2	TFO
2	4-2	Volterra	Volterra	Volterra

Notes: see notes in Table 2

Table 4 shows error rates on the testing data of the best architecture for each variables combination with simple moving average calculation. Table 4 shows the accuracy of MLFF and GMDH results decrease in long term prediction. Meanwhile the results of GMDH approach are better than MLFF neural network.

Table 4: MLFF & GMDH network results with SMA

Variable Combination	MLFF		GMDH	
	RME	MAPE	RME	MAPE
MA_5,MA_5(-1),MA_5(-2)	1.482	0.789	0.821	0.369
MA_{10},MA_{10}(-1),MA_{10}(-2)	2.283	0.783	1.936	0.621
MA_{50},MA_{50}(-1),MA_{50}(-2)	6.764	1.321	5.273	1.207
MA_{55},MA_{55}(-1),MA_{55}(-2)	8.361	1.702	7.129	1.301

Note: RMSE stands for Root Mean Square Error; and MAPE for Mean Absolute Percentage Error

Table 5 shows error rates of the best architecture for each variables combination with exponential moving average calculation. Table 5 confirms the results of table 4 regarding to the period of prediction and two different approaches.

Table 5: MLFF & GMDH network results with EMA

Variable Combination	MLFF		GMDH	
	RME	MAPE	RME	MAPE
MA_5,MA_5(-1),MA_5(-2)	1.275	0.351	0.721	0.278
MA_{10},MA_{10}(-1),MA_{10}(-2)	1.786	0.412	1.152	0.356
MA_{50},MA_{50}(-1),MA_{50}(-2)	6.215	1.102	4.982	0.985
MA_{55},MA_{55}(-1),MA_{55}(-2)	7.619	1.456	6.683	1.118

Notes: see notes in Table 4

These tables suggest that the GMDH network with two hidden layers is a better architecture. It is also revealed that exponential moving average (EMA) has a better response for the TEPIX time series in this period. The result of the neural network accuracy measure (RMSE) is noticeably decreased with short-term moving like 5 and 10. This confirms the fact there is short-term dependence in TEPIX.

4.2. Comparison of Power Tracking

Table 6 shows error rates of the best architecture for each method and number of correct tracking with its percentage.

Table 6: MLFF & GMDH network results

Method	Number of Correct Tracking	Percentage of Correct Tracking
GMDH	87	87%
MLFF	74	74%

Notes: the number of predictions is 100

As per table No. 6, the Number of correct tracking by using GMDH is 87 out of 100 while by using MLFF it is 74. Also the percentage of correct tracking by using GMDH is 87% while the same by using MLFF is 74%. As a result, tracking by using GMDH method is better and more powerful than MLFF.

4.3. Comparison of Profitability

Trading signals from the two models are used to generate daily profit and loss expressed as percent returns. Table 7 summarizes the returns for each approach.

Table 7: comparison of profitability

Approach	2007	2008	2009	Bear	Bull
Period return GMDH	0.62	0.61	0.63	0.36	0.55
Period return MLFF	0.59	0.58	0.59	0.31	0.52

With comparison period returns, the GMDH approach generates much better overall returns than the MLFF. While the year-to-year performances of the GMDH model are impressive, it is also useful to compare the relative performances of the models in time periods when prices were generally rising (bull market) or generally falling (bear market).

5. Conclusion

This paper used two types of neural networks to prediction of TEPIX. It has been shown that GMDH type neural networks have better results when the associated system is so complex and the underline processes or relationship are not completely understandable or display chaotic properties.

We have used TEPIX time series and its moving average as the input to the neural network and the results show:

- The GMDH has better response in the forecasting, power tracking and profitability relative to MLFF neural network.
- TEPIX has short-term dependence to its history.
- The exponential moving average is a better choice relative to simple moving average, if the moving average techniques are used.

References

[1] Ritchie, J.C., 1996. Fundamental Analysis: a Back-to-the-Basics Investment Guide to Selecting Quality Stocks. Irwin Professional Publishing

[2] Fama E, Schwert W. Asset returns and inflation. Journal of Financial Economics 1977;5:115–46.

[3] Rozeff M. Dividend yields are equity risk premiums. Journal of Portfolio Management 1984;11:68–75.

[4] Keim D, Stambaugh R. Predicting returns in the stock and bond markets. Journal of Financial Economics1986;17:357–90

[5] Murphy, J.J., 1999. Technical Analysis of the Financial Markets: a Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance.

[6] Coulson , D.R., 1987. The Intelligent Investor's Guide to Profiting from Stock Market Inefficiencies. Probus Publishing

[7] .Van Eyden, R.J., 1996. The Application of Neural Networks in the Forecasting of Share Prices. Finance and Technology Publishing

[8] Box, G., Jenkins, G., 1976. Time Series Analysis: Forecasting and Control. Holden Day.

[9] Pesaran, H.M., Timmermann , A., 1994. Forecasting stock returns: an examination of stock market trading in the presence of transaction costs. Journal of Forecasting 13, 335–367.

[10] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986a. Learning internal representations by backpropagating errors. Nature 323 (99), 533–536. Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986b. Learning internal representations by backpropagating errors. In: Rumelhart , D.E., McClelland, J.L., and the PDP Research Group, (Eds.), Parallel Distributed Processing, vol. 1, MIT Press, Cambridge, MA, pp. 318–362 (Chapter 8).

[11] Haykin, S., 1998. Neural Networks: a Comprehensive Foundation, second ed. Prentice-Hall, Englewood Cliffs, NJ

[12] Meraviglia, C. (1996). Models of representation of social mobility and inequality systems: A neural network approach. Quality & Quantity, 30, 231–252.

[13] Shachmurove, Y. (2005). Business applications of emulative neural networks. International Journal of Business, 10, 303–322

[14] Zhang, G. P., & Berardi, V. L. (2001). Time series forecasting with neural network ensembles: An application for exchange rate prediction

[15] Binner, J. M., Bissoondeeal, R. K., Elger, T., Gazely, A. M., & Mullineux, A. W. (2005). A comparison of linear forecasting models and neural networks: An application to Euro inflation and Euro divisia. Applied Economics, 37,

[16] Plasmans, J., Verkooijen, W., & Daniels, H. (1998). Estimating structural exchange rate models by artificial neural networks. Applied Financial Economics , 8, 541–551.665–680.

[17] Qi, M., & Wu, Y. (2003). Nonlinear prediction of exchange rates with monetary fundamentals. Journal of Empirical Finance, 10, 623–640.

[18] Saltoglu, B. (2003). Comparing forecasting ability of parametric and nonparametric methods: An application with Canadian monthly interest rates. Applied Financial Economics, 13, 169–176.
[19] Donaldson, R. G., & Kamstra, M. (1996). Using dividend forecasting models to reject bubbles in asset prices: The case of the crash of 1929. Review of Financial Studies, 9, 333–383.
[20] Black, A. J., & McMillan, D. G. (2004). Nonlinear predictability of value and growth stocks and economic activity. Journal of Business Finance & Accounting, 31, 439–474.
[21] Jasic, T., & Wood, D. (2004). The profitability of daily stock market indices trades based on neural network predictions: Case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965–1999. Applied Financial Economics, 14, 285–297
[22] Kanas, A., & Yannopoulos1, A. (2001). Comparing linear and nonlinear forecasts for stock returns. International Review of Economics & Finance, 10, 383–398
[23] Kanas, A. (2001). Neural network linear forecasts for stock returns. International Journal of Finance & Economics, 13, 317–354
[24] Maasoumi, E. , & Racine, J. (2002). Entropy and predictability of stock market returns. Journal of Econometrics, 107, 291–312
[25] Qi, M. (1999). Nonlinear predictability of stock returns using financial and economic variables. Journal of Business & Economic Statistics, 17, 419–429.
[26] Rapach, D. E., & Wohar, M. E. (2005). Valuation ratios and long-horizon stock price predictability. Journal of Applied Econometrics, 20, 327–344.
[27] Shively, P. A. (2003). The nonlinear dynamics of stock prices. Quarterly Review of Economics and Finance, 13, 505_517
[28] Hamid , S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S&P 500 Index futures prices. Journal of Business Research, 57, 1116–1125.
[29] Dunis, C. L., & Huang, X. (2002). Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. Journal of Forecasting, 13, 317–354.
[30] Lam.m.(2004):neural network techniques for financial performance prediction: integrating fundamental and technical analysis, decision support systems,374.567-581
[31] Nygren.k.(2004):stock prediction-a neural network approach,master s thesis,royal institute of technology.kth,Sweden.
[32] Kasstra.l and boyd,m (1995)designing a neural network for forecasting financial and economic time series.neurocomputing.(10)
[33] Jasic,t and wood,d.(2004)the profitability of daily stock market indices trades based on neural network predictions:case study for the s&p 500. the dax. the TOPIX and the FTSE in the period 1965_1999 applied financial economics,14,285-297
[34] Yao, J., Li, Y., & Tan , C. L. (2000). Option price forecasting using neural networks. Omega, 28, 455–466
[35] Amilon, H. (2003). A neural network versus Black–Scholes: A comparison of pricing and hedging performances. Journal of Forecasting, 22, 317–335.
[36] Malliaris, M., & Salchenberger, L. (1996). Using neural networks to forecast the S& P 100 implied volatility. Neurocomputing, 10, 183–195.
[37] Binner, J. M., Bissoondeeal, R. K., Elger, T., Gazely, A. M., & Mullineux , A. W. (2005). A comparison of linear forecasting models and neural networks: An application to Euro inflation and Euro divisia. Applied Economics, 37, 665–680.
[38] Heston, S. L., & Nandi, S. (2000). A closed-form GARCH option valuation model. Review of Financial Studies, 13, 585–625.
[39] Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. Journal of Finance, 49, 851–859.
[40] Lin, C. T., & Yeh, H. Y. (2005). The valuation of Taiwan stock index option prices – comparison of performances between Black–Scholes and neural network model. Journal of Statistics & Management Systems, 8, 355–367.

[41] Qi, M. (1999). Nonlinear predictability of stock returns using financial and economic variables. Journal of Business & Economic Statistics, 17, 419–429.

[42] Neftci, S.N., Naïve trading rules in financial markets and Wiener-Kolmogorov prediction theory: a study of technical analysis. Journal of Business 64, 549–571, 1991

[43] Brock, W.A., Lakonishok, J., LeBaron, B. Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance 47, 1731–1764,1992

[44] Gencay, Ramazan, Non-linear prediction of security returns with moving average rules. Journal of Forecasting 15 (3), 165–174.1996

[45] Gencay, Ramazan, The predictability of security returns with simple technical trading rules. Journal of Empirical Finance 5 (4), 347–359, 1998a.

[46] Gencay, Ramazan, Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics 47 (1), 91–107, 1999.

[47] Gencay, Ramazan, Stengos, Thanasis, Moving average rules, volume and the predictability of security returns with feedforward networks. Journal of Forecasting, 17 (5–6), 401–414, 1998.

[48] Akgiray, Vedat, Booth, G. Geoffrey, Hatem, John J., & Mustafa, Chowdhury; Conditional dependence in precious metal prices. Financial Review, 26, 367-386, 1991

[49] Kamarthi, S.V. and Pittner, S; Accelerating neural network training using weight extrapolation; Neural Network, 12, 1285-1299, 1999.

[50] Kartalopoulos, S.V.; Understanding Neural Networks and Fuzzy Logic- Basic Concepts and Applications; Prentice Hall, New-Delhi, 2000

[51] Gencay, Ramazan, Stengos, Thanasis, Moving average rules, volume and the predictability of security returns with feedforward networks. Journal of Forecasting, 17 (5–6), 401–414, 1998.

[52] Lippmann, R.P.; An introduction to computing with neural nets; IEEE Mag. 3 (4), P.P: 4-22, 1987

[53] Srinivasan, N., Ravichandran, V., Chan, K.L., Vidhya, J.R., Ramakirishnan, S., Krishnan, S.M.; Exponentiated backpropagation algorithm for multilayer feedforward neural networks; Neural Information Processing, 2002. ICONIP apos;02. Proceedings of the 9th International Conference, Volume 1, Issue, 18-22 Nov. 2002 Page(s): 327 - 331 vol.1

[54] Ivakhnenko A.G., J.-A. Müller; Recent Developments of Self-Organising Modeling in Prediction and Analysis of Stock Market; 1995

[55] Madala,H.R. and Ivakhnenko,A.G.; Inductive Learning Algorithms for Complex Systems Modeling; CRC Press Inc., Boca Raton, 1994, p.384

[56] Farlow, S.J., ed.; Self-organizing Method in Modelling: GMDH type algorithm; Marcel Dekker Inc., 1984

[57] Ivakhnenko A. G., Muller J. A.; Self-organization of nets of active neurons: System Analysis Modeling and Simulation; (SAMS), Vol. 20, No. 1-2, pp.29-50, 1995.

[58] Vandaele W.; Applied time series and Box-Jenkins models; Academic Press, 1983